

Agentes Autónomos e Sistemas Multi-Agente (AASM)

Ano Lectivo 2007/2008

Laboratório 5 - Introdução ao JADE

1 Objectivos

- Introdução à plataforma JADE

2 Hello World

Para se familiarizar com o JADE analise o agente simples que lhe foi fornecido.

- Corra o executável `hello.bat`.
- Analise a saída na consola.
- Analise o conteúdo do ficheiro executável.
- Analise o código do agente (`aasm.jade.HelloWorldAgent` na directoria `source`).
- O JADE fornece várias ferramentas gráficas de suporte. Corra o executável `hello-gui.bat` e explore estas ferramentas.

3 Comportamento

Um agente em JADE pode ter vários comportamentos associados. Um comportamento é definido com base na classe `jade.core.behaviours.Behaviour`. Este classe define dois métodos principais: `action()` e `done()`. O primeiro implementa a acção do comportamento cada vez que este é executado; o segundo indica quando o comportamento termina.

- Adicione ao `HelloWorldAgent` a classe interna `MyFirstBehaviour` que implementa um comportamento simples. Este comportamento termina após a primeira execução e como acção escreve a frase *“Este é um comportamento simples!”* no ecrã. Associe o comportamento ao agente utilizando o método `addBehaviour`.
- Estenda este comportamento de forma a receber no construtor o número de vezes que a frase deve ser escrita. Esse valor deve ser dado ao agente na sua construção modificando o ficheiro `hello.bat`.

O JADE disponibiliza vários tipos de comportamentos, entre outros:

- `CyclicBehaviour` que representa um comportamento infinito que corre em todos os ciclos de execução. Nesta classe o método `done()` é redefinido para devolver sempre `false`.
- `TickerBehaviour` que representa comportamentos periódicos, isto é, que são executados de x em x tempo. Nesta classe o método `done` também é redefinido para devolver sempre `false` e o método `action` é redefinido para executar o método `onTick()`. O intervalo de tempo deve ser indicado no construtor da instância.
- `WakerBehaviour` que representa comportamentos que são executados, uma vez, após um dado período de tempo. Nesta classe o método `done()` é redefinido para devolver sempre `true` e o método `action` é redefinido para executar o método

`onWake()`. O tempo de espera é introduzido no constructor do comportamento. É possível correr o comportamento novamente invocando o método `reset()`.

4 Comunicação

O JADE inclui uma plataforma de comunicação entre agentes. Como linguagem de comunicação é utilizada uma ACL (*Agent Communication Language*) baseada na especificação da FIPA (*Foundation for Intelligent Physical Agents*). Cada mensagem é representada por uma instância da classe `jade.lang.acl.ACLMessage` que contém, entre outras coisas, informação sobre o tipo de mensagem (*performative*), o seu conteúdo, o seu remetente e o seu destinatário. A comunicação é gerida pelos métodos `send()` e `receive()` do agente.

- Implemente um agente `Emissor` que além de escrever a frase “Vou enviar uma mensagem ao *Receptor*” no ecrã, envia uma mensagem ao dito *Receptor*. Note que no construtor da mensagem deve indicar o seu tipo. A título de exemplo utilize `ACLMessage.INFORM`. O nome do receptor deve ser recebido como argumento no construtor do agente (altere o ficheiro `hello.bat`) e deve corresponder ao nome de um agente existente (ou seja, a um agente declarado no mesmo ficheiro).
- Implemente um outro agente `Receptor` que tem um comportamento cíclico que verifica se o agente recebeu alguma mensagem e a imprime no ecrã. O agente deve ficar à espera de uma nova mensagem. Para isso, deve utilizar o métodos `receive()` e `block()`.
- Uma vez implementados estes dois agentes, volte a executar o ficheiro `hello-gui.bat`. Experimente enviar mensagens a um `Receptor` e verifique que elas são recebidas.

No exemplo acima descrito, o agente `Receptor` recebe qualquer tipo de mensagem. No entanto, isso nem sempre é desejado. Para isso, o JADE permite filtrar as mensagens que são recebidas usando instância da classe `jade.lang.acl.MessageTemplate`.

- Adicione ao agente o comportamento que trata apenas de mensagens do tipo `ACLMessage.INFORM`. Para seleccionar as mensagens utilize o método `MatchPerformative` da classe `MessageTemplate`.
- Utilizando a interface gráfica, verifique o novo comportamento do agente apenas trata de mensagens do tipo `INFORM`.
- A linguagem disponibiliza mais filtros: por exemplo, o método `MatchSender` que selecciona as mensagens de acordo com o emissor. Além disso, permite a composição de filtros através dos métodos `and` e `or` da classe `MessageTemplate`.

5 Caso de estudo: Leilões

Conforme estudado nas aulas teóricas, [Wooldridge02, Cap. 7], existem vários tipos de leilões. Neste aula vamos simular dois deles. Para isso, consideram-se as seguintes entidades:

- **Representante**: está no leilão em representação do utilizador que o criou e que lhe dá o valor máximo de oferta e item a comprar no momento em que o cria;
- **Leiloeiro**: coordenador que recebe as propostas e vende os artigos escolhendo a melhor proposta feita. Adicionalmente, indica aos proponentes se a sua proposta foi aceite, ou não. Nos leilões do tipo *inglês*, também é o responsável por avisar os proponentes cujas propostas foram ultrapassadas para que estes as possam cobrir.

a) Leilão do tipo *First-price sealed-bid*

Neste tipo de leilão, todos os interessados fazem a sua primeira (e única) proposta durante um período previamente estabelecido.

Findo esse período, o leiloeiro escolhe a melhor proposta (tipicamente, quem oferece mais) que é paga de acordo com o proposto.

Este tipo de leilão é o utilizado, por exemplo, nos concursos públicos em que a melhor proposta dentro de um dado prazo é aquela que é aceite.

Explore o código de base fornecido e termine a implementação deste leilão.

São fornecidas as seguintes classes (incompletas):

```
aasm.jade.firstPriceSealedBidAuction.Leiloeiro  
aasm.jade.firstPriceSealedBidAuction.Representante
```

b) Páginas Amarelas

Note que na função `obtemLeiloeiros` do código fornecido que o agente `Representante` assume a existência de certos leiloeiros “bem conhecidos”.

O JADE suporta um mecanismo de Páginas Amarelas (*Directory Facilitator*, DF). Explore este mecanismo e altere as classes de modo a que o `Leiloeiro` registre os seus serviços no DF e que o `Representante` os descubram questionando o DF.